

Parallel Computation of Electron–Molecule Collisions

Carl Winstead, Howard Pritchard, and Vincent McKoy
California Institute of Technology

When electrons and molecules collide, interesting—and sometimes useful—results follow. Methods adapted to parallel computers can now predict the outcomes of collisions for rather large molecules, thereby opening a new range of processes to study.

THE STUDY OF COLLISIONS BETWEEN ELECTRONS AND MOLECULES is an old field, almost as old as the discovery of the electron itself. Yet it has never been a large field; indeed, if we confine our interest (as we do here) to low-energy collisions, only a handful of research groups, experimental or theoretical, are active throughout the world today. It might therefore be appropriate to begin with two questions: Why study such collisions at all? And why is so little research being done?

There are both basic and applied answers to the first question. From the purely scientific point of view, low-energy electron–molecule collisions are uniquely interesting in several ways. (See the accompanying sidebar for an explanation of terms.) As manifestations of quantum-mechanical scattering, such collision processes are intermediate in complexity between the potential-scattering problems found in textbooks and the collisions between two heavy particles (for example, two molecules) that are of interest in such areas as chemical reaction dynamics. The electron, being structureless, has no internal degrees of freedom that might be excited or de-excited during the collision, and so it can be characterized completely by its asymptotic velocity and the orientation of its intrinsic spin. The formulation of the scattering problem is thus greatly simplified when one of the collision partners is an electron. Being light, however, the electron has a long de Broglie wavelength and so persists in acting fully quantum-mechanical at much higher energies than would a heavy particle. (The lightest heavy particle found in normal matter, the proton or H^+ ion, is over 1,800 times as massive as the electron; common atoms such as carbon and oxygen are more than 10 times heavier still.) Yet at all but the lowest energies, the electron's wavelength is not long enough, compared to the size of the molecule, to permit simplifying approximations.

Meanwhile, a molecule makes a very interesting collision partner compared to, say, an atom. Atoms can almost always be treated as spherically symmetric, and the only internal degrees of freedom they exhibit (at energies of interest to us) are electronic—that is, changes in the state of motion of

the electron cloud surrounding the nucleus. Molecules, on the other hand, have rotational degrees of freedom (because they aren't round) as well as vibrational degrees of freedom (because they are made up of atoms that can oscillate relative to one another). The possibility of exciting these motions adds an extra level of complexity to electron-molecule collisions. A second level of complexity also derives from the molecule's aspherical shape, namely the dependence of the scattering outcome on both the electron's direction of incidence and its direction of departure relative to the molecule. (Though the molecule is rotating, its rotation is slow enough that we can almost always treat it as fixed in orientation for the duration of the collision.) These complexities must often be neglected or averaged away, but the development of means to measure or calculate the relevant quantities is a topic of abiding research interest.

Where does this problem appear?

Applied interest in electron-molecule collisions derives mainly from the possibility that the molecule may be excited, dissociated, or ionized in such collisions. Examples of systems in which electron-molecule collisions are important include the upper atmosphere, shocks, and electrical discharges—environments far from thermal equilibrium, where sources of energy sufficient to generate electrons by ionization encounter matter in molecular form.

An area of special interest to us over the past several years is low-temperature plasma processing of materials, one of the basic technologies of the semiconductor industry. Reactor designs vary, but a common configuration illustrating general principles is this: a gas mixture flows between two electrodes, at one of which sits the surface to be modified (a semiconductor wafer, for example). A high-frequency electromagnetic field whose peak strength is sufficient to cause electrical breakdown is applied between the electrodes. Electrons accelerated by the field collide with molecules of the gas, generating excited molecules, ions, atoms, and radicals (neutral fragments). Some of the ions, atoms, and radicals so generated reach the surface being treated, where they produce some desired effect—sputtering, etching, deposition of a new layer, and so on. By using polymer or oxide masks to protect selected areas of the surface, patterns can be formed and microelectronic circuits built up. Of course such systems must be carefully controlled to avoid producing undesired effects.

If we were to generate the reactive species that modify the surface thermally—by direct heating of the gas—we would need temperatures of thou-

Terms and Concepts

Collisions and scattering theory

The interaction between sunlight and raindrops that produces a rainbow is a manifestation of electromagnetic scattering. When petroleum seismologists image rock layers by making loud noises and listening closely to the echoes, they are engaged in applied acoustic scattering. In the microscopic world of atoms and molecules, where the wave properties of matter are evident, quantum-mechanical scattering theory is used to interpret—or to predict—what happens when objects with wavelike properties collide. Predicting the rates of gas-phase chemical reactions, for example, is a (very formidable!) problem in quantum scattering theory. Our own interest is the somewhat more tractable problem of electrons colliding with molecules. We use the laws of quantum mechanics to compute probabilities for different outcomes of a collision, such as elastic scattering or excitation of the molecule to a state that then falls apart. In the same way that knowledge of the acoustic properties of different rocks is important to the seismologist, electron-collision probabilities for various gases are important to understanding transport and chemistry in fields such as atmospheric physics and plasma reactor design.

Collision energies

When we say we study low-energy electron-molecule collisions, what is meant by "low energy"? A convenient energy unit is the electron-volt (eV), the kinetic energy an electron picks up when accelerated through a one-volt potential drop (1.6×10^{-19} joules, in SI units). The kinetic energies of the outer or valence electrons of molecules and atoms, those electrons responsible for chemical bonding, are on the order of 10 eV. The low-energy collision regime, in our usage, is that in which the incident electron's kinetic energy is comparable to that of a valence electron—roughly 0 to 100 eV. At higher energies, approximations that greatly simplify the collision calculation begin to be valid.

Cross sections

Collision probabilities are conventionally expressed as effective areas, or cross sections. In classical collision physics the connection between cross sections and probabilities is simple. For example, the probability that a dart thrown across a room will hit a soap bubble is proportional, at low bubble densities, to the distance the dart flies, the density of bubbles, and the (average) cross-sectional area of one bubble. In quantum-mechanical scattering, we retain the name *cross section* for the analogous proportionality constant, with units of area, connecting the number of collision events having a specified outcome to the projectile's path length and the density of targets. Cross sections defined in this way may bear little relation to the "true" size of the target and will depend on the identity of the projectile, the collision energy, and what outcome is specified—including the direction of scattering and the energy transfer, if any, between projectile and target.

sands or tens of thousands of kelvins. Even if the gas density and exposure times were small enough to avoid outright melting of the surface, the resulting thermal damage would be unacceptable. The secret of low-temperature plasma processing



lies in oscillating the applied field. The frequency of this oscillation (typically 13.56 MHz) is high enough that only the electrons, and not the much heavier ions, acquire substantial kinetic energies before the field reverses direction. Kinematics prevents the transfer of significant kinetic energy in an electron-molecule collision just as it does in the collision between a ping-pong ball and a bowling ball; thus, the average kinetic energy of the heavy particles (that is, their temperature) remains low. Electron-molecule collisions, then, are an essential mechanism in low-temperature plasmas, uniquely able to transfer energy to molecules' internal degrees of freedom while avoiding energy transfer to translational degrees of freedom.

More sophisticated and more reliable plasma models and reactor simulations are being developed. Tools of this type will be needed in the future, since the growth in density and complexity of microelectronic circuitry places ever greater demands on the plasma processes used.¹ Reliable models will require, as input, extensive and accurate data about the microscopic processes that occur in the plasma and at the surface, with electron-molecule collision data being particularly important.

Why are such data not, in general, available? The short answer is that these things are both hard to measure and hard to calculate. Measurements are difficult for a number of reasons:

- ◆ low-energy electrons are easily deflected by stray fields,
- ◆ inelastic processes often have small cross sections (see the sidebar) and overlap in energy, and
- ◆ many of the gases used in industry are hazardous to both humans and laboratory equipment.

Calculations are difficult because the underlying equation, the Schrödinger equation, is a second-order differential equation in many variables. Even if we neglect the motion of the molecule's nuclei, as we often can, we must consider its electrons on an equal footing with the projectile electron. Even for small molecules this can mean dozens of variables. Approximation methods must be used, but they must be methods capable of high accuracy. Such methods are very demanding of computational resources, and progress in calculating cross sections for polyatomic molecules has awaited the development of high-performance computing. Those interested in details of our technique for solving this problem can read about it in the next section, which might otherwise be skimmed; the succeeding section deals with computational issues.

Computing cross sections

Rather than try to compute the wave function (which contains all of the physical information) from scratch, we choose a limited "space" or set of known functions, and look for the best approximate wave function that can be formed as a combination of functions in that space. To do this, we must give a precise meaning to the term "best approximate wave function." This can be done by means of a variational principle: an expression for some quantity of interest that is stationary about the exact answer, or in other words contains only second-order errors when the wave function itself contains first-order errors. Since we are interested in scattering, we use a variational principle for the scattering amplitude, whose square modulus gives the cross section or probability.

The starting point is Schrödinger's equation, $\hat{H}\Psi=0$, where $\hat{H}=E-H$ is the energy minus the Hamiltonian operator and $\Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N+1})$ is the $(N+1)$ -electron wave function for an N -electron molecule plus an incident electron. Dividing H into the Hamiltonian for a free molecule and free electron, H_0 , and an interaction term V , we can rewrite Schrödinger's equation as

$$(E - H_0)\Psi = V\Psi \quad (1)$$

and formally "solve" to obtain the Lippmann-Schwinger integral equation:

$$\Psi = \psi + G_0^{(+)}V\Psi \quad (2)$$

Here ψ is a solution to $(E - H_0)\psi = 0$, and $G_0^{(+)}$ is the Green's function associated with H_0 . That is,

$$G_0^{(+)}(E) = \lim_{\epsilon \rightarrow 0^+} \frac{1}{E - H_0 + i\epsilon} \quad (3)$$

The quantity we are after is the scattering amplitude $f_m(\mathbf{k}, \mathbf{k}')$: this describes the transition from an initial situation in which the electron has a momentum (in appropriate units) \mathbf{k} and the molecule is in a state labeled n to a final situation with electron momentum \mathbf{k}' and target state n' . Using the Lippmann-Schwinger equation, the scattering amplitude can be written in a number of different ways; the particular form we are interested in is

$$\begin{aligned} -2\pi f_m(\mathbf{k}, \mathbf{k}') = & \langle \psi_{n'}(\mathbf{k}') | V | \Psi_n(\mathbf{k}) \rangle \\ & + \langle \Psi_{n'}(\mathbf{k}') | V | \psi_n(\mathbf{k}) \rangle \\ & - \langle \Psi_{n'}(\mathbf{k}') | V - V G_0^{(+)} V | \Psi_n(\mathbf{k}) \rangle \end{aligned} \quad (4)$$

(Here we use Dirac bra-ket notation, in which $\langle |$ and $| \rangle$ angle brackets imply integration over the electron coordinates $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N+1}$.) By applying

Equation 2, it is easy to show that this expression amounts to $f = f + f - f$. The reason we are interested in this peculiar form is that, as Julian Schwinger first pointed out,² it is variationally stable—that is, first-order errors in the wave functions $\Psi_n(\mathbf{k})$ and $\Psi_n(\mathbf{k}')$ lead to second-order errors in the computed approximation to $f_{nn}(\mathbf{k}, \mathbf{k}')$. As is true of the Rayleigh-Ritz procedure commonly applied to the eigenvalue problems of computational chemistry, the variational stability of Schwinger's expression is the key to the formulation of an efficient approximation method.

To implement Equation 4 numerically, we introduce a basis set: that is, we approximate $\Psi_n(\mathbf{k})$ and $\Psi_n(\mathbf{k}')$ as sums of known functions $\chi_i(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N+1})$. We determine the unknown coefficients $x_i^{(n)}$ and $y_i^{(n')}$ (where the superscripts (n) and (n') are indices, not exponents) in these sums by imposing the requirement of variational stability:

$$\frac{\partial f}{\partial x_i^{(n)}} = \frac{\partial f}{\partial y_i^{(n')}} = 0 \quad (5)$$

These steps lead to working equations

$$A\mathbf{x}^{(n)} = \mathbf{b}^{(n)}, \quad \mathbf{y}^{(n')\dagger} A = \mathbf{b}^{(n')\dagger} \quad (6)$$

where the dagger indicates the Hermitian conjugate and the elements of matrices A and \mathbf{b} (a one-column matrix, or vector) are given by

$$A_{ij} = \langle \chi_i | V - VG_0^{(+)}V | \chi_j \rangle \quad (7a)$$

and

$$b_i^{(n)} = \langle \chi_i | V | \psi_n(\mathbf{k}) \rangle \quad (8)$$

For problems involving only one channel, or state n of the target, these working equations are adequate. When more than one channel is involved, however, representation of the Green's function $G_0^{(+)}$ can be problematic. For this reason, we work with a slight modification³ of Equation 4 in which a projection operator P is introduced to project $G_0^{(+)}$ onto a finite number of energetically accessible channels. This modification preserves variational stability and leads to the replacement of Equation 7a with

$$A_{ij} = \langle \chi_i | \left(\frac{1}{N+1} - P \right) \hat{H} + PV - VG_p^{(+)}V | \chi_j \rangle \quad (7b)$$

where $G_p^{(+)}$ is the projected version of $G_0^{(+)}$. An important feature of the resulting expression is that the trial wave function need not satisfy scattering boundary conditions³ and may therefore be expanded in numerically convenient functions such as Cartesian Gaussians.

The last step in reducing the Schwinger variational principle to practice is to express the many-electron functions χ_i and $\psi_n(\mathbf{k})$ in such a way that we can evaluate the matrix elements A_{ij} and $b_i^{(n)}$ of Equations 7 and 8. The χ_i we take to be spin-adapted determinants of one-electron orbitals—configuration state functions (CSFs), in quantum-chemical terminology. The orbitals are in turn represented as sums of Cartesian Gaussian functions ζ_α ,

$$\begin{aligned} \zeta_\alpha(\mathbf{r}; \mathbf{R}, l, m, n) \\ = N_\alpha (x-X)^l (y-Y)^m (z-Z)^n \exp(-\alpha |\mathbf{r} - \mathbf{R}|^2) \end{aligned} \quad (9)$$

The $\psi_n(\mathbf{k})$ are products of target electronic states Φ_n and plane waves $\exp(i\mathbf{k} \cdot \mathbf{r}_{N+1})$ that represent a free electron; by also approximating Φ_n as a sum of CSFs, we at last reduce all needed integrals in Equations 7 and 8 to three types. The first type includes one- and two-electron Coulomb-potential integrals over Gaussians,

$$\left\langle \zeta_\alpha(\mathbf{r}) \left| \frac{1}{|\mathbf{r} - \mathbf{R}_\alpha|} \right| \zeta_\beta(\mathbf{r}) \right\rangle \quad (10)$$

and

$$\left\langle \zeta_\alpha(\mathbf{r}_1) \zeta_\beta(\mathbf{r}_2) \left| \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \right| \zeta_\gamma(\mathbf{r}_1) \zeta_\delta(\mathbf{r}_2) \right\rangle \quad (11)$$

as well as one-electron kinetic-energy integrals

$$\left\langle \zeta_\alpha(\mathbf{r}) \left| \nabla^2 \right| \zeta_\beta(\mathbf{r}) \right\rangle \quad (12)$$

where ∇^2 is the Laplacian operator in \mathbf{r} . The second type is similar to the first, but one of the Gaussians is replaced by a plane wave, $\exp(i\mathbf{k} \cdot \mathbf{r}_{N+1})$, giving one-electron integrals

$$\left\langle \zeta_\alpha(\mathbf{r}) \left| \frac{1}{|\mathbf{r} - \mathbf{R}_\alpha|} \right| \exp(i\mathbf{k} \cdot \mathbf{r}) \right\rangle \quad (13)$$

and two-electron integrals

$$\left\langle \zeta_\alpha(\mathbf{r}_1) \zeta_\beta(\mathbf{r}_2) \left| \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \right| \zeta_\gamma(\mathbf{r}_1) \exp(i\mathbf{k} \cdot \mathbf{r}_2) \right\rangle \quad (14)$$

No kinetic-energy integrals of this second type are required. Both types of integrals can be evaluated analytically (which in fact is why Gaussians are used instead of the physically more appropriate exponential functions). The remaining type of integral involves the Green's function. Matrix elements of $VG_p^{(+)}V$ cannot be evaluated analytically; however, an efficient quadrature procedure can be devised⁴ that requires only integrals of the second type.

Electron-molecule collision calculations are done on a single molecule, usually taken to be fixed in space. Most experimental situations, and most natural or technological environments where electron-molecule collisions occur, involve

signed without scalability in mind are unlikely to be successful on MPPs.

Program organization

Our program (known as SMC, for “Schwinger multichannel”) for treating electron–molecule scattering performs each of the significant computational tasks—integral evaluation, transformation of integrals to required matrix elements, I/O, and solution of the linear equations—in parallel. We wrote the program using a coarse-grained, loosely synchronous message-passing model that is appropriate to distributed-memory MPPs based on powerful microprocessors. Its original implementation was for the JPL/Caltech Mark III^{fp} hypercube; since then, we have ported it to MPPs made by Intel, nCube, and Cray Research.

It is easy to see that the integral evaluation step in our calculation can be parallelized. Each integral can be computed independently of the others from the parameters defining the Gaussian and plane-wave functions involved. We need only provide each processor with the relevant parameters, routines for computing an integral, and a mechanism for deciding which subset of integrals to compute. This mechanism must be chosen very carefully, however, since it will affect almost every other aspect of the parallel program, including load balance, scalability, and the amount of communication overhead.

Cross-section calculations for larger molecules may require 10^{13} floating-point operations, pointing up the need for high-performance computers.

cause the quadrature of $VG_p^{(v)}V$ is over the magnitude and angles of a \mathbf{k} variable.) When the number of channels is small, most of the time is spent evaluating integrals, partly because the number of operations required to evaluate one integral is fairly large and partly because the transformation step is more efficient than the evaluation step (for reasons explained later). As C increases, however, the transformation of the integrals rapidly comes to dominate. The I/O associated with the quadrature data set (whose volume grows as C^3G^2) and the solution of the system $A\mathbf{x} = \mathbf{b}$ (which requires on the order of C^3G^3 operations) also become significant parts of the calculation when C is large.

Data decomposition

With this partitioning scheme, each processor will have the same number of integrals to evalu-

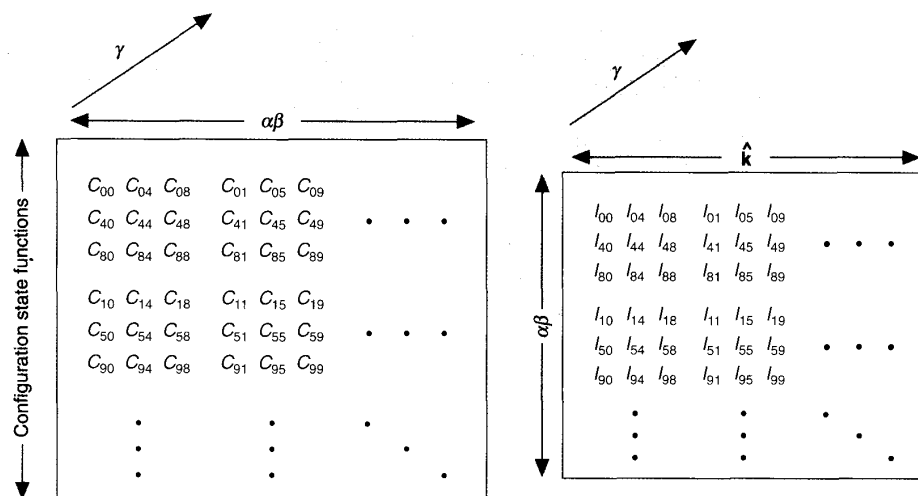


Figure 1. Scheme for partitioning the calculation of integrals and performing integral transformations in parallel. The right-hand matrix contains integrals $I(\alpha, \beta, \gamma, \mathbf{k})$ defined by Equation 14 in the text. The row index labels (α, β) pairs, and the column index labels angles \mathbf{k} . Both dimensions are distributed in scattered fashion over processors. The integrals are transformed into matrix elements involving many-electron configuration state functions through distributed matrix multiplication using a coefficient matrix C (on the left). The process is repeated for each γ .

ate, but the time required to evaluate each integral can vary depending on the parameters of the functions involved. To avoid load imbalance, we use a scattered, rather than block, distribution of the (α, β) -pair and \mathbf{k} indices: successive indices are assigned to successive rows or columns of the processor mesh in cyclic fashion, as shown in Figure 1. This "scrambling" of assignments, together with the large number of integrals (typically thousands) each processor computes for fixed γ and $|\mathbf{k}|$, achieves good load balance.

Communication

Load balance is one essential attribute of an efficient parallel program; low communication overhead is another. Computing the integrals requires no communication at all, so we are off to a good start. Yet the integrals are of no interest in themselves: we are after the matrices A and $b^{(n)}$. Many steps intervene between the integrals and the elements of these matrices. These steps include linear transformations from the Gaussian basis to a different set of one-electron functions (called *molecular orbitals*), application of rules for forming many-electron matrix elements from one- and two-electron integrals, and quadratures. Each of these steps necessarily involves communication. Can that communication be done efficiently?

The linear-transformation and quadrature operations pose no difficulties. Both can be cast in terms of matrix multiplication, with a regular pat-

tern of data access, and hence of communication, and with a favorable ratio of computation to communication. The questionable step is the imposition of the rules for constructing many-particle integrals from the one- and two-electron integrals. In our original program, these rules were applied after the linear transformations to molecular orbitals, at which point they connect each integral to at most a few many-electron matrix elements. Accomplishing this sparse transformation on a parallel machine would appear to require either massive data motion with relatively little computation, a very intricate program controlling a correspondingly intricate communication pattern, or both. Suppose, however, we fuse this step with the linear-transformation step preceding it. Then the rules governing the formation of many-electron matrix elements, rather than being applied directly, can be used to control the construction of a coefficient matrix. This will take us, in one operation, from the integrals of Equations 13 and 14 to the desired many-particle matrix elements. The elements of this coefficient matrix are labeled in one dimension by pairs (α, β) of Gaussians, and in the other dimension by the many-electron functions (see Figure 1); they are formed by summing products of the coefficients for the linear transformation from Gaussians to molecular orbitals in appropriately weighted combinations.⁵ The coefficient matrix can be constructed without interprocessor communication,

Our program spends the great majority of its time cycling through the steps just described, constructing and transforming successive batches of integrals. Periodically, as all integrals for a given magnitude $|\mathbf{k}|$ are completed, quadratures over the angles $\hat{\mathbf{k}}$ are done (also by distributed matrix multiplication) and intermediate data are written

singular systems is needed, we save the \mathbf{A} and $\mathbf{b}^{(n)}$ matrices to disk and later apply an SVD (singular-value decomposition) solver on a sequential machine. Solving the equations in either case is a minor constituent of the overall computation.

Our program has been used in production work on a number of parallel machines, including the nCube 2, the Cray T3D, and the Intel iPSC/860, Delta, and Paragon. Though we have not engaged in detailed benchmarking, we have paid attention to performance issues and made fairly extensive optimizations where warranted. The computers we used vary greatly in speed and each has its idiosyncrasies, but certain general observations about our program's performance apply to all machines and may therefore be of interest to those contemplating the parallelization of similar programs.

or more floating-point operations per clock tick without using special vector registers, *in practice* the main memory in current MPPs cannot provide operands and store results fast enough to sustain such speeds. Memory limitations can often be reduced with careful instruction scheduling but usually at the expense of writing assembler code. Truly breaking the memory bottleneck requires reuse of data in fast (cache) memory, so that the ratio of floating-point operations to main-memory references is high. Not all algorithms can be implemented to reuse cache this way, and compilers do not always do an adequate job with those that can. The user who, like us, is not willing to resort to assembly programming therefore may expect a large fraction of peak performance only when able to employ optimized library routines for common procedures like matrix multiplications and Fourier transforms.

With the preceding observation as context, it is easy to understand the relative performance of the principal sections of our program. Integrals over Gaussians and plane waves are computed via a fairly intricate set of Fortran subroutines, within which few opportunities for vectorization exist. This part of the computation tends to run at roughly 10 percent of peak both on vector supercomputers and on MPPs. In the construction of the transformation matrix, there is some complicated logic at the top level, but most of the computational work is in the form of vector-vector manipulations. Optimized routines from the BLAS 1 library can be used here, but with no opportunity for cache reuse; therefore performance, though it may be two or three times better than for integral evaluation, is still far below peak. The transformation of integrals to many-electron matrix elements, however, being a distributed-matrix multiplication, has as its kernel the multiplication of local matrix blocks. Matrix-matrix multiplication does very well on current MPPs because of the extensive cache reuse possible; on large problems 90 percent or more of peak speed can be obtained. Thus, in spite of the communication overhead it contains, the transformation step is by far the most efficient of our major procedures.

The relative importance of the different sections of our program varies considerably depending on the molecule under study and the type of investigation being done. Consequently, the overall speed also varies considerably. It would of course be possible to design a calculation that was entirely dominated by integral transformations and that would display very impressive perfor-

[illegible]

mance, but we have not tried to do this. In the course of our production work so far, we have seen sustained speeds up to about 5 Gflops for some runs on larger systems such as a 256-processor T3D and a 512-processor Paragon; we expect to see considerably higher speeds on more extensive calculations to be undertaken in the near future. The impact of parallel computing on our work can be gauged from the fact that speeds in the 5-Gflops range represent an improvement of about two orders of magnitude over what our program achieved on vector supercomputers.

I/O and communication

So far we have discussed only floating-point performance. Interprocessor communication and disk I/O are also important considerations on MPPs. Since most of our communication occurs inside a distributed-matrix multiplication, communication overhead is not significant unless the problem is small enough, and the computer large enough, that the matrix subblocks assigned to individual processors are very small. (How small is "very small" depends on the communication bandwidth and latency of the MPP.) In that case, however, we ought simply to run on fewer processors. Disk I/O is a more serious consideration. Most of our I/O is associated with the quadrature data set for evaluating the Green's function. In some cases, the size of this data set can reach a few gigabytes. Both its writing and its reading can be parallelized, since each processor stores and subsequently retrieves different data. However, we found that in one case (the Intel Paragon), seemingly natural parallel I/O strategies may perform at much less than 1 Mbyte per second, implying I/O times for multigigabyte data sets on the order of hours. That much overhead is tolerable, if barely, while generating the data set, since its generation requires hours of computation and it need be written only once. However, we typically wish to read it back many times in order to calculate cross sections at many different energies. Fortunately, restructuring the data set and the way it is read to match the physical structure of the parallel file system gave us vastly improved performance—about 30 Mbytes per second, close to the maximum expected rate on the particular Paragon used.

Applications

We have applied our method to low-energy electron collisions with a variety of molecules over the past several years. As mentioned earlier, a particular focus of our research has been the computation of cross sections relevant to modeling low-

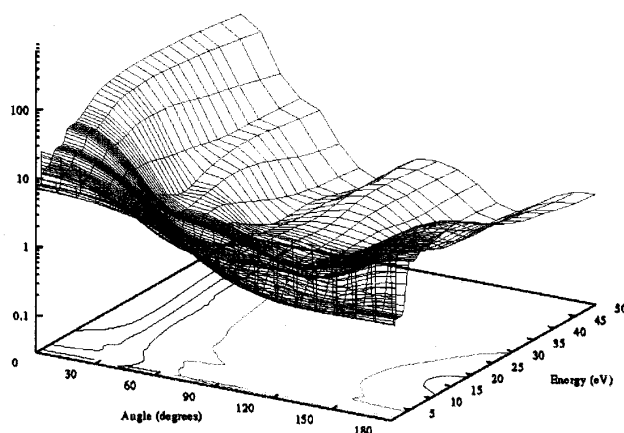


Figure 2. Cross section for the elastic scattering of low-energy electrons by boron trichloride, BCl_3 .

temperature plasmas used in semiconductor manufacturing. We have looked at

- ◆ hydrides such as PH_3 , SiH_4 , and Si_2H_6 used in doping or deposition;
- ◆ halogen compounds such as CF_4 and BCl_3 used in etching; and
- ◆ species such as SiCl and SiF_4 that are formed as byproducts of etching.

Many of our recent studies have been carried out as part of a Sematech-funded project to generate cross-section data for BCl_3 etching plasmas. In addition to these semiconductor-related studies, we have pursued a long-standing interest in atmospheric species with investigations of molecules such as N_2 , CO , and CO_2 .

Figure 2 illustrates some typical features of low-energy electron-molecule cross sections. In particular it shows a calculated cross section for elastic scattering of electrons by BCl_3 as a function of both the collision energy and the scattering angle. At low energy, the cross section displays considerable structure. This structure is due to *resonances*, that is, states in which the impinging electron becomes temporarily bound to the molecule, forming a metastable negative ion. At collision energies where resonances occur, the probability of scattering is enhanced, and peaks are observed in the cross section as a function of energy. The angular pattern of scattering is also affected by resonances, since the probability of escape in various directions reflects the symmetry of the state in which the electron is temporarily trapped. At higher energies, the cross section becomes smoother as a function of both energy and angle and begins to decrease in magnitude; small-



angle or near-forward scattering and large-angle or near-backward scattering (the latter from "hard" or nearly head-on collisions) come to dominate. These features are typical of elastic cross sections, though the details, especially the number and nature of resonances, vary. Inelastic collisions have many of the same features, but generally have much smaller cross sections and a greater variety of angular behavior.

As parallel computers continue their rapid advance in size and power, we anticipate that substantial improvements will be possible in both the scope and accuracy of computational studies of electron-molecule collisions. At the same time, refinements and enhancements to our programs will make it possible to study processes that are currently beyond our capabilities. The future of computational electron-molecule collision research thus appears bright.

The greatest promise of high-performance computing, however, lies not in isolated advances in individual fields of study such as ours, but in the simultaneous development of computational approaches to a whole set of closely related problems. In the case of plasma-based materials processing, for example, there is reason to hope that

data from numerical studies of collisions and reactions will support sophisticated, three-dimensional plasma models and that those models can in turn form the basis of computer-aided design and optimization tools for plasma reactors. Such a complete computational treatment of a complex physical system, extending from the submicroscopic realm (where quantum mechanics applies), to the macroscopic scale of eight-inch wafers, would have been unimaginable, or at least unimaginably expensive, without the dramatic advances in computer power that MPPs based on microprocessors have made possible. We look forward to making a useful contribution as such unified computational approaches evolve. ♦

Acknowledgments

We gratefully acknowledge support of this research by the National Science Foundation under the Grand Challenge project "Parallel I/O Methodologies for I/O-Intensive Grand Challenge Applications," by the Air Force Office of Scientific Research, and by Sematech, Inc.

References

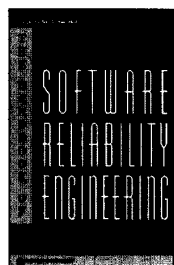

1. National Research Council, *Plasma Processing of Materials: Scientific Opportunities and Technological Challenges*, National Academy Press, Washington, D.C., 1991.
2. J. Schwinger, "A Variational Principle for Scattering Problems," *Physical Review*, Vol. 72, 1947, p. 742.
3. K. Takatsuka and V. McKoy, "Extension of the Schwinger Variational Principle Beyond the Static-Exchange Approximation," *Physical Rev. A*, Vol. 24, 1981, p. 2,473-2,480.
4. M.A.P. Lima et al., "Applications of the Schwinger Multichannel Method to Electron-Molecule Collisions," *Physical Rev. A*, Vol. 41, 1990, pp. 327-332.
5. C. Winstead and V. McKoy, "Studies of Electron-Molecule Collisions on Massively Parallel Computers," in *Modern Electronic Structure Theory*, Vol. 2, D.R. Yarkony, ed., World Scientific, Singapore, 1995.

Carl Winstead is a senior research fellow at the California Institute of Technology. His research interests include electron-molecule scattering, molecular photoionization, and the development of high-performance computational methods for solving quantum-mechanical problems. He has a BA in chemistry from Rice University and a PhD in physical chemistry from Indiana University. He is a member of the American Physical Society.

Howard Pritchard is a programmer/analyst in the Applications Division at Cray Research. He received his PhD in chemical physics from California Institute of Technology and a BA from Rice University. He is a member of the American Physical Society.

Vincent McKoy is professor of theoretical chemistry at the California Institute of Technology. His research interests include theoretical and computational studies of molecular photoelectron spectroscopy and of collisions of low-energy electrons with molecules. He received his BS in chemical engineering from the Technical University of Nova Scotia and his PhD in chemistry from Yale University. He is a member of the American Physical Society.

The authors can be reached in care of Carl Winstead, A.A. Noyes Laboratory of Chemical Physics, California Institute of Technology, Pasadena, CA 91125; e-mail, clw@cco.caltech.edu.




Software Reliability Engineering Handbook

by Michael R. Lyu

Contains the best current software reliability engineering practices in various industries, including telecommunications, computer, avionics, defense and finance. This book/disk set is a definitive guide to today's most-used reliability techniques, models, tools, and solutions. It covers measurement and prediction, the effects of product and development process metrics on operational software behavior, and the application of this knowledge in specifying and guiding software development, acquisition, use, and maintenance.

850 pages. January 1996. Hardcover. ISBN 0-07-039400-8.
Catalog # RS00030 — \$66.50 Members / \$69.50 List



IEEE
COMPUTER SOCIETY

